



The Islamic University of Gaza
Faculty of Engineering
Computer Engineering Department



Introduction to Computers

LAB #3

Elementary Programming 2

Eng. Hashem M. Hejazy

Eng. Mohammed O. AlHabbash

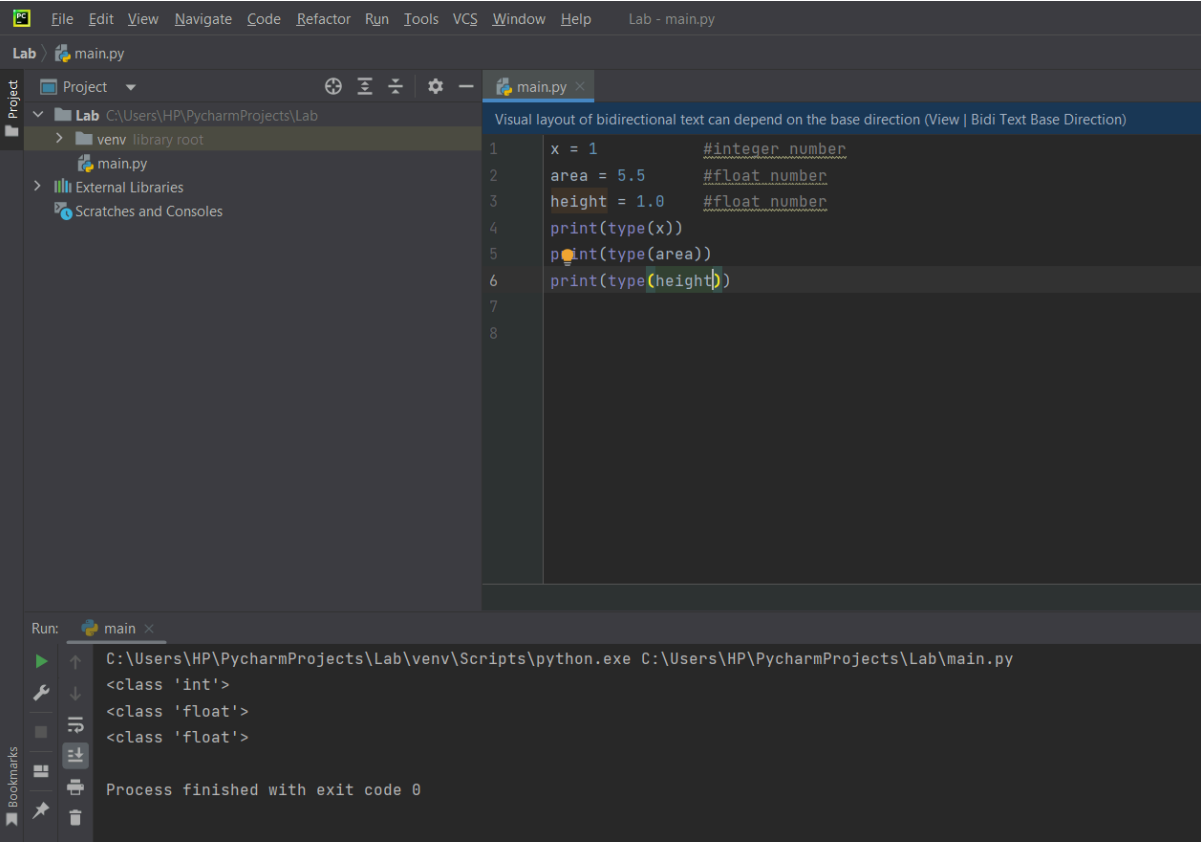
Spring 2026

Python data types

- Booleans are either True or False.
- Numbers can be integers (1 and 2), floats (1.1 and 1.2), fractions (1/2 and 2/3), or even complex numbers (i, the square root of -1).
- Strings are sequences of Unicode characters, e.g., an HTML document.
- Bytes and byte arrays, e.g., a JPEG image file.
- Lists are ordered sequences of values.
- Tuples are ordered, immutable sequences of values.
- Sets are unordered bags of values.
- Dictionaries are unordered bags of key-value pairs.

Numbers

Python supports both integers and floating point numbers. Integer types are for representing whole numbers, for example 7, 55 or 3652. floating types are for representing numbers with a fractional part, for example 7.7, 3.14 or 111.637. There's no type declaration to distinguish them; Python tells them apart by the presence or absence of a decimal point.



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the Project view with a tree structure: Lab (C:\Users\HP\PycharmProjects\Lab) containing a venv directory (library root) and a main.py file. The main editor window displays the code in main.py:

```
1 x = 1 #integer number
2 area = 5.5 #float number
3 height = 1.0 #float number
4 print(type(x))
5 print(type(area))
6 print(type(height))
7
8
```

Below the editor, the Run window shows the execution output for the main.py script:

```
Run: main
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
<class 'int'>
<class 'float'>
<class 'float'>
Process finished with exit code 0
```

Arithmetic operators

Python provides operators, which are special symbols that represent computations like addition and multiplication.

Operator	Example
+ Addition	$2 + 5 = 7$, $2.5 + 3.2 = 5.7$, $1 + 2.0 = 3.0$
- Subtraction	$6 - 1 = 5$
* Multiplication	$3 * 3.2 = 9.6000000000000001$
/ Division	$10.2 / 5.1 = 2.0$
% Modulus	$5 \% 2 = 1$
** Exponent	$3 ** 2 = 9$
// Floor Division	$7 // 2 = 3$

Notes

1. You can use the `type()` function to check the type of any value or variable.
2. Adding an int to an int yields an int.
3. Adding an int to a float yields a float. Python coerces the int into a float to perform the addition, then returns a float as the result.

Order of operations

When an expression contains more than one operator, the order of evaluation depends on the order of operations. For mathematical operators, Python follows mathematical convention. The acronym PEMDAS is a useful way to remember the rules:

- Parentheses have the highest precedence and can be used to force an expression to evaluate in the order you want. Since expressions in parentheses are evaluated first, $2 * (3-1)$ is 4, and $(1+1)**(5-2)$ is 8. You can also use parentheses to make an expression easier to read, as in $(\text{minute} * 100) / 60$, even if it doesn't change the result.

- Exponentiation has the next highest precedence, so $1 + 2^{**}3$ is 9, not 27, and $2 * 3^{**}2$ is 18, not 36.
- Multiplication and Division have higher precedence than Addition and Subtraction. So $2*3-1$ is 5, not 4, and $6+4/2$ is 8, not 5.
- Operators with the same precedence are evaluated from left to right (except exponentiation). So in the expression $\text{degrees} / 2 * \text{pi}$, the division happens first and the result is multiplied by pi. To divide by 2π , you can use parentheses or write $\text{degrees} / 2 / \text{pi}$.

Augmented Assignment Operators

The operators $+$, $-$, $*$, $/$, $\%$, and $**$ can be combined with the assignment operator ($=$) to form augmented assignment operators.

Operator	Example
$+=$ Addition assignment	$x += 1 \Rightarrow x = x + 1$
$-=$ Subtraction assignment	$x -= 1 \Rightarrow x = x - 1$
$*=$ Multiplication assignment	$x *= 1 \Rightarrow x = x * 1$
$/=$ Division assignment	$x /= 1 \Rightarrow x = x / 1$
$\%=$ Remainder assignment	$x \% = 1 \Rightarrow x = x \% 1$
$**=$ Exponent assignment	$x ** = 1 \Rightarrow x = x ** 1$

Python Strings

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks either pairs of single or double quotes.
- Subsets of strings can be taken using the slice operator ($[]$ and $[:]$) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end So $[-1]$ represent last index).
- The plus ($+$) sign is the string concatenation operator.
- The $*$ operator also works on strings; it performs repetition. For example, $\text{'Spam'} * 3$ is 'SpamSpamSpam' . If one of the values is a string, the other has to be an integer.
- The $\text{len}(\text{string_name})$ used to calculate the length of the string

```
1 str = 'This Is Lab 3'
2 print(str) # Prints complete string
3 print(str[0]) # Prints first character of the string
4 print(str[2: 5]) # Prints characters starting from 3rd to 5th
5 print(str[4:]) # Prints string starting from 3rd character
6 print(str * 3) # Prints string two times
7 print(str + "Exam") # Prints concatenated string
8 print("length str " + len(str))
```

Run: main x

```
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
This Is Lab 3
T
Is
Is Lab 3
This Is Lab 3This Is Lab 3This Is Lab 3
This Is Lab 3Exam
length str 13
Process finished with exit code 0
```

String concatenation

which means it joins the strings by linking them end-to-end.

The + operator concatenates strings to create a new string which has both strings as ordered.

```
1 firstName='Mohammed'
2 secondName='Ali'
3 fullName=firstName+secondName
4 print('Your Full Name Is ' + firstName + secondName)
5 print('Your Full Name Is ' + firstName + ' ' + secondName)
6
```

Run: main x

```
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
Your Full Name Is MohammedAli
Your Full Name Is Mohammed Ali
Process finished with exit code 0
```

Reversing a Python String

With Accessing Characters from a string, we can also reverse them. We can Reverse a string by writing `[::-1]` and the string will be reversed.

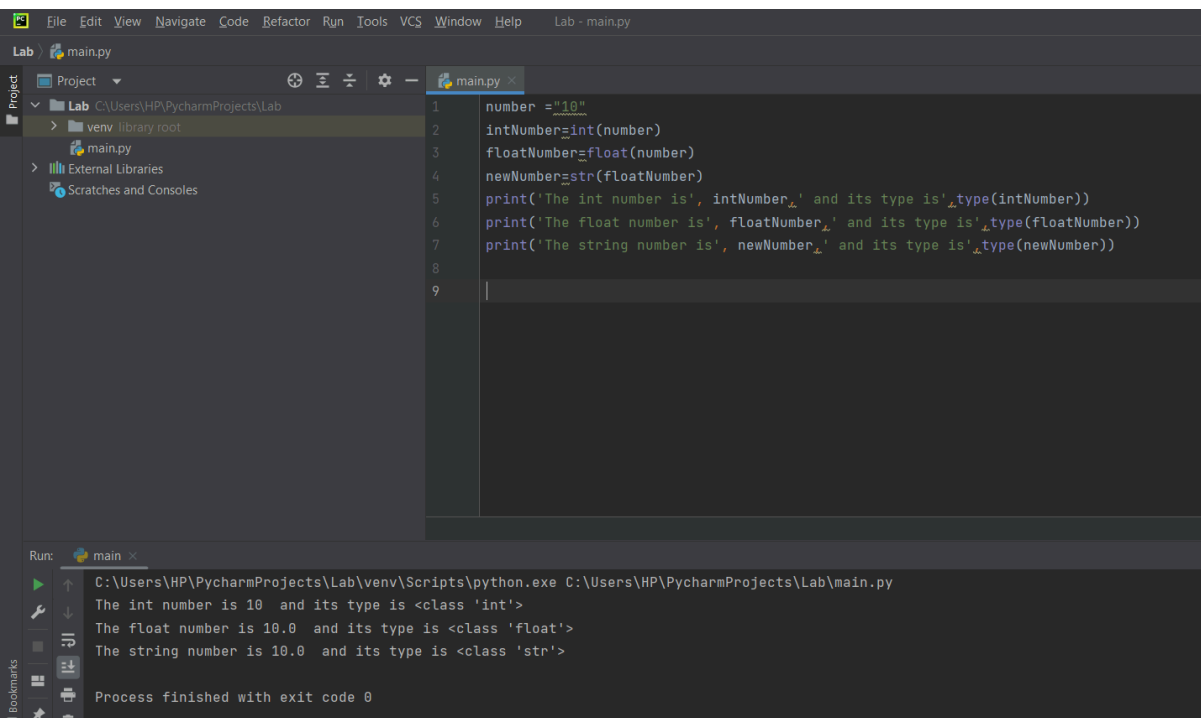
```
firstName='Mohammed'  
print(firstName[::-1])
```

Output

```
demmahOM
```

Type Conversions and Rounding

The datatypes we studied so far are integer, float and string. We can convert between those data types using the suitable method.



```
1 number = "10"  
2 intNumber = int(number)  
3 floatNumber = float(number)  
4 newNumber = str(floatNumber)  
5 print('The int number is', intNumber, ' and its type is', type(intNumber))  
6 print('The float number is', floatNumber, ' and its type is', type(floatNumber))  
7 print('The string number is', newNumber, ' and its type is', type(newNumber))  
8  
9
```

Run: main ×
C:\Users\HP\PycharmProjects\Lab\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Lab\main.py
The int number is 10 and its type is <class 'int'>
The float number is 10.0 and its type is <class 'float'>
The string number is 10.0 and its type is <class 'str'>
Process finished with exit code 0

